Amendments to the Claims:

1. (currently amended) A method for remotely managing a computing device, comprising:

receiving, by the computing device, a service request sent by a remote application via an out-of-band (OOB) connection;

storing the service request in a selected storage location;

polling the selected storage location by a an operating system resident multiplexing agent for new requests;

determining by the multiplexing agent a subagent corresponding to the received service request;

invoking the corresponding subagent on the computing device determined to correspond to the service request, wherein the corresponding subagent corresponding to the service request services the service request; and

sending a response to the remote application to indicate that the service request has been performed.

2. (currently amended) The method as recited in claim 1, wherein the determined subagent determined to correspond to the service request is a system management basic input output system (SMBIOS) agent, and wherein the SMBIOS agent accesses the SMBIOS tables to fulfill the service request.

3. (original) The method as recited in claim 1, wherein the selected storage location is a receive message queue (RMQ) construct of intelligent platform management interface (IPMI).

4. (original) The method as recited in claim 3, wherein the service request comprises header information identifying a client sending the service request.

5. (currently amended) The method as recited in claim 4, wherein the response is sent to the ~~identified~~ client <u>identified as having sent the service request</u> using a send message construct of IPMI.

6. (original) The method as recited in claim 1, wherein the subagent registers a callback function with the multiplexing agent, wherein the callback function corresponds to a service request type.

7. (currently amended) The method as recited in claim 6, wherein ~~a~~ <u>the</u> subagent has a plurality of corresponding callback functions.

8. (original) The method as recited in claim 1, wherein the multiplexing agent continues to poll the selected storage location simultaneously with the servicing of a service request by the subagent.

9. (original) The method as recited in claim 1, further comprising accepting dynamic updates of available subagents by the multiplexing agent.

10. (original) The method as recited in claim 9, wherein accepting dynamic updates of available subagents comprises:

identifying an added dynamic link library in a predetermined storage location, the added dynamic link library corresponding to a new subagent; and

registering at least one callback function corresponding to the added dynamic link library with the multiplexing agent, wherein the identifying and registering are performed during runtime.

11. (currently amended) A machine accessible <u>storage</u> medium ~~comprising~~ <u>having</u> instructions <u>stored thereon</u> for servicing out-of-band service requests <u>in a computing device</u>, the instructions ~~structured~~ <u>when executed on a machine</u> ~~to~~ cause <u>the</u> ~~a~~ machine to:

receive, by the computing device, a service request sent by a remote application via an out-of-band (OOB) connection;

store the service request in s a selected storage location;

poll the selected storage location by a an operating system resident multiplexing agent for new newly received service requests;

determine by the multiplexing agent a subagent corresponding to the received service request;

invoke the corresponding subagent on the computing device determined to correspond to the service request, wherein the corresponding subagent corresponding to the service request services the service request; and

send a response to the remote application to indicate that the service request has been performed.

12. (currently amended)   The machine accessible storage medium as recited in claim 11, wherein the determined subagent determined to correspond to the service request is a system management basic input output system (SMBIOS) agent, and wherein the SMBIOS agent accesses the SMBIOS tables to fulfill the service request.

13. (currently amended)   The machine accessible storage medium as recited in claim 11, wherein the selected storage location is a receive message queue (RMQ) construct of intelligent platform management interface (IPMI).

14. (currently amended)   The machine accessible storage medium as recited in claim 13, wherein the service request comprises header information identifying a client sending the service request.

15. (currently amended)   The machine accessible storage medium as recited in claim 14, wherein the response is sent to the identified client identified as having sent the service request using a send message construct of IPMI.

5

16. (currently amended)   The machine accessible storage medium as recited in claim 11, wherein the instructions are structured to register a callback function with the multiplexing agent, by the subagent, wherein the callback function corresponds to a service request type.

17. (currently amended)   The machine accessible storage medium as recited in claim 16, wherein a subagent has a plurality of corresponding callback functions.

18. (currently amended)   The machine accessible storage medium as recited in claim 11, wherein the multiplexing agent continues to poll the selected storage location simultaneously with the servicing of a service request by the subagent.

19. (currently amended)   The machine accessible storage medium as recited in claim 11, further comprising instructions structured to accept dynamic updates of available subagents by the multiplexing agent.

20. (currently amended)   The machine accessible storage medium as recited in claim 19, wherein instructions structured to accept dynamic updates of available subagents comprise:

identifying an added dynamic link library in a predetermined storage location, the added dynamic link library corresponding to a new subagent; and

registering at least one callback function corresponding to the added dynamic link library with the multiplexing agent, wherein the identifying and registering are performed during runtime.

21. (currently amended)   A system for servicing out-of-band (OOB) service requests, comprising:

a processor communicatively coupled to a memory store and a baseboard management controller (BMC), wherein the BMC is configured to accept service requests from a remote application communicating with the BMC via an OOB connection, wherein accepted service requests are stored in a selected storage location in the memory store;

a̶ an operating system resident multiplexing agent running on the processor, the multiplexing agent polling the selected storage location for a̶ ̶n̶e̶w̶ newly received service requests r̶e̶q̶u̶e̶s̶t̶; and

at least one subagent running on the processor, wherein a subagent corresponding to a service request type is invoked by the multiplexing agent in response to receiving a n̶e̶w̶ service request.

22. (original) The system as recited in claim 21, wherein one of the at least one subagent is a system management basic input output system (SMBIOS) subagent, wherein the SMBIOS subagent services requests requiring access to SMBIOS tables.

23. (original) The system as recited in claim 21, wherein the selected storage location is a receive message queue (RMQ) construct of intelligent platform management interface (IPMI).

24. (original) The system as recited in claim 23, wherein the service request comprises header information identifying a client sending the service request.

25. (currently amended)   The system as recited in claim 24, wherein a response is sent to the i̶d̶e̶n̶t̶i̶f̶i̶e̶d̶ client identified as having sent the service request using a send message construct of IPMI to indicate service request completion.

26. (original) The system as recited in claim 21, wherein the at least one subagent registers a callback function with the multiplexing agent, wherein the callback function corresponds to a service request type.

27. (original) The system as recited in claim 26, wherein a subagent has a plurality of corresponding callback functions.